

aus

Oliver J. Bott, Peter Fricke, Uta Priss, Michael Striewe (Hrsg.)

Automatisierte Bewertung in der Programmierausbildung

Digitale Medien in der Hochschullehre Band 6

2017, 420 Seiten, br., 42,90 €, ISBN 978-3-8309-3606-0



Waxmann Verlag GmbH

www.waxmann.com info@waxmann.com

7 Automatisierte Bewertung im Kontext der App-Entwicklung am Beispiel Android

Britta Herres, Rainer Oechsle und David Schuster

Zusammenfassung

In diesem Kapitel wird das ASB-System (Automatische Softwarebewertung) der Hochschule Trier beschrieben, das seit mehreren Jahren für unterschiedliche Programmierlehrveranstaltungen eingesetzt wird. In den letzten Jahren wurde das System erweitert, um auch Android-Apps automatisch bewerten zu können. Aus Sicht der Studierenden ist die Einreichung einer Android-App und das Betrachten der Ergebnisse der Bewertungsmaßnahmen nicht wesentlich anders als für andere Programme. Wir beschreiben in diesem Kapitel, wie das ASB-System im Rahmen der unterschiedlichen Lehrveranstaltungen eingesetzt wird, wie die Studierenden und die Dozenten das System benutzen, welche Art von Vorgaben in den Programmieraufgaben vorkommen und welche Erfahrungen wir im Umgang mit dem System gemacht haben.

7.1 Einleitung

Im Fachbereich Informatik der Hochschule Trier wird seit 2006 das webbasierte System ASB (Automatische Softwarebewertung) zur Bewertung der studentischen Lösungen von Programmieraufgaben entwickelt und eingesetzt. Das ASB-System ermöglicht Dozenten, Programmieraufgaben zu ihren Vorlesungen zu erstellen, zu denen Studierende Lösungen hochladen können. Die von den Studierenden programmierten Lösungen werden durch mehrere Bewertungsmaßnahmen automatisch überprüft.

Neben einer Überprüfung der Einhaltung von Programmierkonventionen (z. B. Einrückungen) werden die studentischen Programme ausgeführt. Dabei wird getestet, ob die Programme sich so wie vorgegeben verhalten. Die Ergebnisse der

Bewertungen werden den Studierenden angezeigt. Studierende können ihre Programme daraufhin ändern und erneut zur Überprüfung einreichen, solange die Abgabefrist noch nicht abgelaufen ist. Das System wird seit mehreren Jahren für die Übungen mehrerer Module sowohl im Präsenz- als auch im Fernstudium eingesetzt. Eine frühe Version des ASB-Systems wurde in [Mor+07] beschrieben. Momentan arbeiten wir mit der Version 4, auf der dieser Beitrag basiert.

Relativ neu ist die Nutzung der automatischen Programmbewertung für das Bachelormodul *Entwicklung mobiler Anwendungen*, das im Sommersemester 2013 erstmalig angeboten wurde. Als Programmierplattform wird in diesem Modul Android verwendet. Um auch für dieses Modul die Lösungen von Programmieraufgaben automatisch bewerten zu können, wurde das ASB-System so erweitert, dass nun auch Studierende ihre in den Übungen entwickelten Android-Apps auf das ASB-System hochladen können und entsprechendes Feedback dazu bekommen. In diesem Beitrag wird der Einsatz des ASB-Systems im Allgemeinen und speziell für das Modul *Entwicklung mobiler Anwendungen* beschrieben.

7.2 Einsatzszenarien

Das ASB-System wird im Fachbereich Informatik der Hochschule Trier sowohl im Präsenz- als auch im Fernstudium genutzt. Im Präsenzstudium wird es momentan in vier und im Fernstudium in zwei Modulen eingesetzt. Im Folgenden wird die Einbettung des ASB-Systems in die Lehre etwas allgemeiner und unabhängig von den konkreten Modulen beschrieben, wobei jedoch zwischen dem Einsatz im Präsenz- und Fernstudium unterschieden wird.

Die Module des Präsenzstudiums, in denen ASB genutzt wird, bestehen wie die meisten Module der Informatik-Bachelorstudiengänge aus je einer wöchentlich abgehaltenen 90-minütigen Vorlesung (= 2 SWS [Semesterwochenstunden]) und einer wöchentlich durchgeführten 90-minütigen Übungsveranstaltung (= 2 SWS). Die Übungen werden in Gruppen mit jeweils ca. 20 – 30 Studierenden durchgeführt, wobei sich die Anzahl der Gruppen nach der voraussichtlichen Gesamtzahl der teilnehmenden Studierenden richtet. Jedes dieser Module wird mit 5 ECTS-Punkten kreditiert. Als Lernplattform für das Präsenzstudium wird an der Hochschule Trier *Stud.IP*¹ eingesetzt. Über *Stud.IP* werden den Studierenden Lehrmaterialien wie Vorlesungsunterlagen, Beispielprogramme und Übungsaufgaben von den Dozenten bereitgestellt. Außerdem wird das Forum von *Stud.IP* genutzt. Die Forumnutzung ist mal mehr, mal weniger intensiv.

1 <http://www.studip.de>

In der Regel wird jede Woche nach der Vorlesung ein Übungsblatt auf *Stud.IP* veröffentlicht, das die Studierenden bis zur Übungsstunde der folgenden Woche bearbeiten müssen. In den Übungsstunden werden die Aufgaben von den Studierenden dem betreuenden Dozenten sowie den anderen anwesenden Studierenden der Gruppe vorgestellt und diskutiert, wobei die Studierenden sich teils dazu freiwillig melden können, teils werden sie auch zufällig aufgerufen. Studierende, die zu häufig die Aufforderung, ihre Lösung vorzustellen, verweigern, können die Prüfungsvorleistung, die zur Teilnahme an der schriftlichen 90-minütigen Prüfung berechtigt, im laufenden Semester nicht mehr erwerben. Zu Beginn des Semesters wird bekannt gegeben, wie häufig eine solche Aufforderung ohne Konsequenzen abgelehnt werden darf.

Einige der Aufgaben der veröffentlichten Übungsblätter sind als sogenannte *ASB-Aufgaben* gekennzeichnet. Das heißt, hierzu muss bis zu einer angegebenen Frist eine Lösung auf den ASB-Server hochgeladen werden. Die Studierenden können relativ schnell die automatischen Bewertungsergebnisse zu ihren hochgeladenen Lösungen einsehen. Sollte der ASB-Server anzeigen, dass Mängel festgestellt wurden, können die Studierenden ihre Software verändern und die so veränderte Software erneut auf den ASB-Server laden. Es gibt keine Versuchszählung, sodass dies beliebig oft möglich ist, bis die Abgabefrist erreicht ist. Die Abgabefrist liegt in der Regel jede Woche immer einige Stunden vor Beginn der Übungsstunde der ersten Gruppe, sodass die Übungsbetreuenden noch die Möglichkeit haben, die finalen Bewertungsergebnisse der eingereichten Programme vor Beginn der Übungsstunden einzusehen. Auch für die ASB-Aufgaben gibt es bestimmte Mindestanforderungen, die zu Beginn des Semesters mitgeteilt werden und die die Studierenden erfüllen müssen, um die Prüfungsvorleistung zu erwerben. Die Anforderungen sind nicht derart, dass alle Einreichungen der Studierenden komplett mängelfrei sein müssen. Es wird stattdessen gefordert, dass für einen Großteil der gestellten ASB-Aufgaben ein ernsthaftes Bemühen zur Lösung der Aufgabe erkennbar ist. Ob es ein solches „ernsthaftes Bemühen“ gibt oder nicht, kann nicht vollautomatisch vom ASB-System festgestellt werden. Hierzu begutachten die Übungsbetreuenden die Lösungen, wobei hier insbesondere die Lösungen in Augenschein genommen werden, die nach Ablauf der Abgabefrist noch Mängel aufweisen. Nach einer kurzen Anlaufzeit wird von den Studierenden erwartet, dass die formalen Anforderungen, die an ihre Programme gestellt werden (z. B. die richtige Struktur der Ordner und Dateien in der hochgeladenen ZIP-Datei oder die korrekte Formatierung der Quellcodedateien), eingehalten werden. Sollten diese eher einfach zu erfüllenden Anforderungen wiederholt nicht erfüllt werden, führt dies zum Verlust der Prüfungsvorleistung.

Der Anteil derjenigen Aufgaben, die als *ASB-Aufgaben* gekennzeichnet sind, ist im Präsenzstudium je nach Modul unterschiedlich. Dies hängt im Wesentlichen davon ab, ob für die unterschiedlichen Aufgabenstellungen überhaupt Tests entwickelt werden können, wie aufwändig dies gegebenenfalls ist und wie schnell die Tests entwickelt werden können. Im Moment wird ASB nur in Modulen eingesetzt, in denen Java als Programmiersprache verwendet wird. Obwohl geplant war, den Einsatz auch auf Module, die Python oder C++ nutzen, auszudehnen, ist dies bis jetzt nicht geschehen. Es folgt eine Übersicht über die einzelnen Module des Präsenzstudiums, in denen ASB eingesetzt wird:

- Modul *Grafische Benutzeroberflächen*: Der Anteil an ASB-Aufgaben für dieses Modul war vergleichsweise hoch. Mit der Umstellung von Swing auf JavaFX müssen allerdings alle Tests neu entwickelt werden, wobei zunächst geklärt werden musste, wie JavaFX-Programme grundsätzlich getestet werden können. Hierbei war die besondere Herausforderung, herauszufinden, ob und wie es möglich ist, JavaFX-Programme im sogenannten Headless-Modus (d. h. ohne einen an den Rechner angeschlossenen Bildschirm) zu testen. Dies ist eine wichtige Voraussetzung, da der ASB-Server auf einer virtuellen Maschine ohne Bildschirm betrieben wird. Nachdem diese Schwierigkeit überwunden wurde, beginnt nun die Entwicklung neuer Testfälle für die ASB-Aufgaben. In den Testfällen wird die Benutzerinteraktion (Eingaben über Tastatur und Maus) mit dem zu testenden Programm über eine Programmierschnittstelle simuliert. Anschließend wird überprüft, ob Elemente der Benutzeroberfläche bestimmte Eigenschaften aufweisen (z. B. ob in einem Anzeigenfeld ein bestimmter Text steht oder ob eine Schaltfläche eine bestimmte Farbe hat). Um einen größeren Nachdruck auf die Bearbeitung der Übungsaufgaben zu legen, müssen die Studierenden ihre Lösungen auch zu manchen Aufgaben auf den ASB-Server laden, obwohl es dafür (noch) keine Tests gibt. Hier werden dann lediglich formale Überprüfungen durchgeführt. Die Übungsbetreuer haben somit aber die Möglichkeit, stichprobenhaft die eingereichten Lösungen durchzusehen.
- Modul *Parallele Programmierung*: Das Testen paralleler Programme ist eine besondere Herausforderung, da sich der genaue Ablauf nicht steuern lässt und sich somit Synchronisationsfehler nicht bei jeder Ausführung des Programms manifestieren. Entsprechend ist hier die Abdeckung mit ASB-Aufgaben eher gering. Wie für das Modul *Grafische Benutzeroberflächen* beschrieben, gibt es auch hier ASB-Aufgaben ohne Tests. Für einige wenige Aufgaben werden statische Codeüberprüfungen vorgenommen.

- Modul *Entwicklung verteilter Anwendungen*: In diesem Modul geht es um die Entwicklung von Client-Server-Anwendungen über Sockets und RMI. Ferner wird die Entwicklung webbasierter Anwendungen, die über Servlets und JSF (Java Server Faces) realisiert werden, behandelt. Für die Sockets- und RMI-Aufgaben wird der Client-Code der Studierenden getestet, indem der Testcode einen Server simuliert, der Server-Code der Studierenden wird getestet, indem die Testfälle als Client agieren. Für die webbasierten Anwendungen läuft der Code der Studierenden in einer Umgebung, die einen Webserver nachahmt. Über eine Programmierschnittstelle kann die Testsoftware mit dem ausgeführten Code der Studierenden interagieren. Das Testen ist in diesem Modul für eine Reihe von Fällen gut möglich, sodass wir hier auf eine mittlere Abdeckung mit ASB-Aufgaben kommen.
- Modul *Entwicklung mobiler Anwendungen*: Wenn auch eine Android-App in Java programmiert ist, so kann sie dennoch nicht in einer normalen Java-Laufzeitumgebung ausgeführt werden, sondern nur auf realen Android-Geräten oder innerhalb von Android-Emulatoren. Es musste also zunächst die Möglichkeit geschaffen werden, Android-Apps auf dem ASB-Server mithilfe von Emulatoren ausführen und testen zu können, was eine gewisse Herausforderung darstellte. Nachdem dies nun grundsätzlich möglich ist (vgl. Kapitel 16), liegen die ersten Testfälle vor und können genutzt werden. Im Prinzip funktionieren die Testfälle wie für das Modul *Grafische Benutzeroberflächen*, indem Benutzereingaben simuliert werden und das Vorhandensein bestimmter Eigenschaften in den Elementen der Oberfläche überprüft wird. Die Abdeckung der Aufgaben mit Testfällen ist momentan noch relativ gering, soll aber in Zukunft erhöht werden.

Nachdem damit die Einbettung der automatischen Programmbewertung in den Lehrbetrieb für das Präsenzstudium beschrieben wurde, soll nun der Einsatz im Fernstudium erläutert werden. Das Informatikfernstudium der Hochschule Trier ist ein weiterbildendes Masterstudium, das sich an Quereinsteiger und Quereinsteigerinnen in die Informatik wendet. Damit ist gemeint, dass als Voraussetzung zur Zulassung in der Regel ein Bachelorabschluss gefordert wird in einem Fach, das sich genügend deutlich von der Informatik unterscheidet wie zum Beispiel Maschinenbau oder Elektrotechnik. Durch das gebührenpflichtige und in der Regel berufsbegleitend durchgeführte Fernstudium erhalten die Studierenden eine Zusatzqualifikation. Die Bezeichnung dieses Studiengangs führt aus diesem Grund den Klammerzusatz *Aufbaustudium*.

Im Fernstudium wird das ASB-System in zwei Modulen eingesetzt, die wie fast alle anderen Module des Fernstudiums mit 10 ECTS-Punkten kreditiert werden. Die Vorlesung wird im Fernstudium generell durch ein Selbststudium ersetzt,

wobei hierfür je nach Modul unterschiedliche Arten von Lehrmaterialien zur Verfügung gestellt werden: speziell für das Fernstudium entwickelte Lehrmaterialien (in Papierform und als PDF-Datei bereitgestellt), Bücher, webbasierte Lehrmaterialien sowie Vorlesungsaufzeichnungen. Dazu gibt es im Verlauf des Semesters pro Modul mehrmals abendliche Tutorien, die als Videokonferenz durchgeführt werden. Im Verlauf dieser Tutorien werden exemplarisch Lehrinhalte vertieft, wobei die teilnehmenden Studierenden hierzu direkt Fragen stellen können. Eine Beratung per E-Mail ist ebenfalls jederzeit möglich. Die Übungen bestehen aus Einsendeaufgaben und einem einwöchigen Präsenzpraktikum. Wenn von den Einsendeaufgaben mindestens 50 % erfolgreich bearbeitet wurden, erfolgt eine Zulassung zum einwöchigen Präsenzpraktikum, das mit einer schriftlichen oder mündlichen Prüfung am letzten Tag des Praktikums abgeschlossen wird. In den Modulen *Einführung in die objektorientierte Programmierung* und *Fortgeschrittene Programmieretechniken (FOPT)* wird zur Kontrolle der Einsendeaufgaben das ASB-System eingesetzt, wobei es sich im Modul *FOPT* bei allen Einsendeaufgaben um ASB-Aufgaben handelt. In beiden Modulen ist die verwendete Programmiersprache auch wieder Java. Abschließend sei noch erwähnt, dass das ASB-System auch in einem anderen Fachbereich der Hochschule Trier eingesetzt wird, nämlich im Fachbereich Wirtschaft für die Java-Programmierausbildung des Bachelorstudiengangs Wirtschaftsinformatik.

7.3 Das Modul *Entwicklung mobiler Anwendungen*

Im Folgenden soll näher auf das Modul *Entwicklung mobiler Anwendungen (EMA)* eingegangen werden, da die Nutzung des ASB-Systems in diesem Modul eine besondere Herausforderung darstellt. Bei diesem Modul handelt es sich um ein Pflichtmodul des Bachelorstudiengangs „Informatik – Sichere und mobile Systeme“, der vom Fachbereich Informatik der Hochschule Trier seit 2008 (damals noch unter der Studiengangsbezeichnung „Informatik – Internetbasierte Systeme“) angeboten wird. Der Studienplan sieht vor, dass dieses Modul im 4. Semester zu absolvieren ist. Das Modul *EMA* kann von den Studierenden der drei anderen Informatik-Bachelorstudiengänge („Informatik“, „Informatik – Digitale Medien und Spiele“ und „Medizininformatik“) als Wahlpflichtmodul belegt werden. Das Modul erfreut sich großer Beliebtheit. Es wird jedes Jahr in der Regel von ca. 80 Studierenden belegt, sodass hierfür drei Übungsgruppen benötigt werden. Mehrere Personen treten als Lehrende dieses Moduls auf: die Vorlesung teilen sich 3 – 4 Dozenten im Verlauf der Vorlesungszeit eines Semesters auf, auch die Übungsgruppen werden von mehreren Personen betreut.

Die Studierenden lernen in diesem Modul unter anderem, welche Interaktionselemente Android zur Gestaltung von Oberflächen anbietet (Buttons, TextViews, EditTexts, Menüs, Dialoge usw.), wie die Bildschirminhalte sogenannter Activities über XML-Layout-Dateien spezifiziert werden, wie Reaktionen auf Benutzeraktionen programmiert werden, wie Activities der eigenen App und unterschiedlicher Apps über sogenannte Intents gekoppelt werden, wie die Problematik einer länger dauernden Ereignisbehandlung (z. B. das Herunterladen größerer Dateien) aufgrund einer Benutzerinteraktion (z. B. eines Button-Klicks) gelöst werden kann und wie unterschiedliche Arten von Services (gestartete Services und gebundene Services) zur Realisierung von Aktionen „im Hintergrund“ (d. h. ohne Benutzeroberfläche) eingesetzt werden können. Als wesentliches Lernziel wird im Modulhandbuch beschrieben, dass die Studierenden die Fähigkeit erlernen, selbstständig kleinere bis mittelgroße Android-Anwendungen zu spezifizieren und mithilfe der Entwicklungsumgebung Android Studio umzusetzen.

7.4 Benutzung des ASB-Systems aus Studierendensicht

Das ASB-System ist nach erfolgreicher Anmeldung mit der Hochschulkennung für jeden Studierenden der Hochschule Trier über einen Browser erreichbar. Die Nutzerin in der Rolle Student erhält auf der Startseite eine Übersicht über die Lehrveranstaltungen, zu denen sie Lösungen einreichen kann, sowie eine Auflistung offener Aufgaben (d. h. Aufgaben, zu denen aktuell Lösungen eingereicht werden können). Abbildung 7.1 zeigt die Startseite des ASB-Systems nach erfolgreicher Anmeldung. Wie in Abschnitt 7.1 erwähnt wurde, besitzt jede Auf-

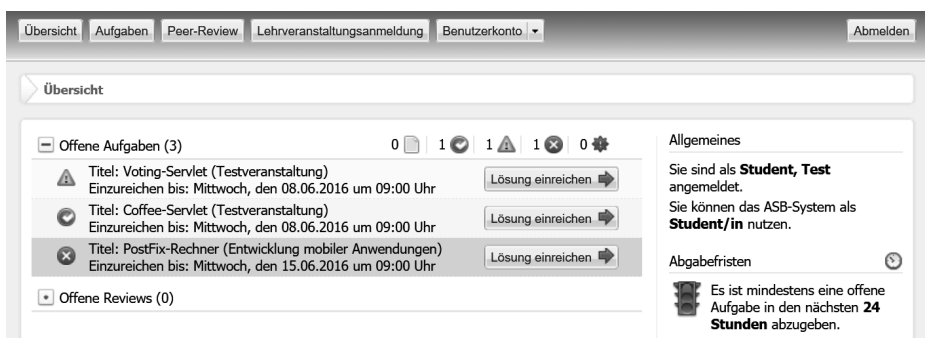


Abbildung 7.1: Studentische Startseite des ASB-Systems

gabe einen Zeitraum, in dem Studierende Lösungen zu dieser Aufgabe hochladen können. Wird diese Abgabefrist überschritten, erscheint die Aufgabe auf der Übersichtsseite nicht mehr. In der Regel wird jede eingereichte studentische Lösung mehreren Bewertungsmaßnahmen unterworfen. Jede einzelne Bewertungsmaßnahme hat pro Aufgabe und pro registriertem Studierenden einen von fünf möglichen Zuständen:

- Noch keine Lösung eingereicht,
- Bewertungsmaßnahme fehlerfrei abgeschlossen (d. h. die Bewertungsmaßnahme konnte erfolgreich durchgeführt werden, dabei gab es keine Beanstandungen),
- Bewertungsmaßnahme mit Warnung abgeschlossen (d. h. die Bewertungsmaßnahme konnte erfolgreich durchgeführt werden, hat aber Punkte gefunden, die noch verbessert werden können wie zum Beispiel die Deklaration einer lokalen Variablen, die im Code gar nicht verwendet wird und somit entbehrlich ist),
- Bewertungsmaßnahme mit Fehlern abgeschlossen (d. h. die Bewertungsmaßnahme konnte erfolgreich durchgeführt werden, hat aber Fehler im zu bewertenden Programm entdeckt),
- Bewertungsmaßnahme fehlgeschlagen (d. h. die Bewertungsmaßnahme konnte nicht durchgeführt werden).

Die Ergebnisse der einzelnen Bewertungsmaßnahmen für eine eingereichte Lösung werden zu einem Gesamtergebnis zusammengefasst. Dabei ist das Gesamtergebnis das Ergebnis einer Einzelbewertungsmaßnahme, das in der obigen Liste am weitesten unten steht. Wenn also beispielsweise ein eingereichtes studentisches Programm durch drei Maßnahmen bewertet wird, wobei eine fehlerfrei, eine mit Warnungen und eine mit Fehlern abgeschlossen wird, so ist das Gesamtergebnis: mit Fehlern abgeschlossen. Als Ergebnis einer Bewertung werden also keine Punkte wie beispielsweise bei einer Klausur vergeben.

Ungeachtet des Ergebnisses der Bewertungsmaßnahmen können die Studierenden bis zum Erreichen der Abgabefrist ihre Lösungen beliebig oft hochladen und bewerten lassen. Allerdings wird für den Zustand der Aufgabe lediglich die letzte Bewertung betrachtet. Auf der Webseite zum Hochladen einer Lösung wird den Studierenden auch eine Übersicht über die Bewertung der zuletzt hochgeladenen Lösung angezeigt, falls zu dieser Aufgabe überhaupt schon etwas eingereicht wurde. Diese Übersicht zeigt an, welche Bewertungsmaßnahmen für die Aufgabe

durchgeführt wurden und zu welchem Ergebnis dies jeweils geführt hat. Abbildung 7.2 zeigt die Bewertungsübersicht nach dem Einreichen einer Lösung.

Die Studierenden sollen bei dieser Aufgabe einen Postfix-Taschenrechner als Android-App implementieren. Die Lösungen zu dieser Aufgabe durchlaufen drei Bewertungsmaßnahmen: *Android*, *ECJ* und *Checkstyle*. Die *Android*-Bewertungsmaßnahme prüft, ob sich die hochgeladene App so verhält, wie es von der Aufgabenstellung gefordert wird. Dazu wird die Anwendung ausgeführt und getestet. Die detaillierte Bewertungsübersicht zeigt in einer Baumstruktur, welche Tests mit Erfolg beziehungsweise nicht erfolgreich abgeschlossen wurden. In der Abbildung ist zu sehen, dass die Funktionalitätstests zum Teil fehlgeschlagen sind. Damit ist das Gesamtergebnis dieser Bewertungsmaßnahme, dass Fehler gefunden wurden. Durch die Auswahl eines fehlgeschlagenen Tests erhalten die Studierenden alle verfügbaren Informationen über den durchgeführten Test, über das

Übersicht | Aufgaben | Peer-Review | Lehrveranstaltungsanmeldung | Benutzerkonto | Abmelden

Übersicht | Aufgaben | Ergebnisse

Aufgabe: PostFix-Rechner (PostFix, Entwicklung mobiler Anwendungen (EMA))
 Lösung: postfix.zip | Eingereicht: 07.06.2016, 09:11:12 Uhr

Neu einreichen:

Bewertung mit Fehlern abgeschlossen

Bewertungsübersicht | Quelltext | **Android** | Checkstyle | ECJ

Detailliertes Bewertungsergebnis | Standardausgabe | Fehlerausgabe

- Ergebnisse des JUnit-Tests
 - Layouttests
 - Funktionalitätstests der Klasse `PostFixCalculator`
 - Prüfen, ob das Eingabefeld geleert wird. **X**
 - Prüfen, ob die zuletzt eingegebene Zahl korrekt gelöscht wird. **X**
 - Prüfen, ob die Division durch Null durch Fehler behandelt wird. **X**
 - Prüfen, ob die Addition zweier beliebiger Zahlen korrekt ausgeführt wird. **✓**
 - Prüfen, ob die Subtraktion zweier beliebiger Zahlen korrekt ausgeführt wird. **✓**
- Funktionalitätstests für die Klasse `PostFixCalculator`

Dieser Testabschnitt prüft, ob sich Ihre Klasse `PostFixCalculator` so verhält, wie es von der Aufgabenstellung wird.
- Prüfen, ob das Eingabefeld geleert wird. **X**

Test aufgrund eines gefundenen Fehlers fehlgeschlagen!

Fehler In diesem Testfall wurde überprüft, ob das Eingabefeld durch Klicken auf den Button DELETE geleert wird. Nach Eingabe der Zahlen "123" und abschließendem Klick auf DELETE wurde ein leeres Eingabefeld erwartet. Ihr Programm zeigte jedoch den Text "123" im Eingabefeld an.

Abbildung 7.2: Studentische Bewertungsübersicht einer eingereichten Lösung am ASB-System

Ist- und das Soll-Verhalten der Anwendung sowie über alle erzeugten Warnungs- und Fehlermeldungen. In diesem konkreten Fall hat der Studierende unter anderem die Funktion des DELETE-Buttons nicht korrekt implementiert, sodass das Eingabefeld nicht geleert wird.

Bei der zweiten Bewertungsmaßnahme *ECJ* wurde der Code mit dem Eclipse-Compiler übersetzt. Abbildung 7.2 zeigt, dass diese Bewertungsmaßnahme erfolgreich war. Gleiches gilt auch für die dritte Bewertungsmaßnahme *Checkstyle*². Diese prüft den eingereichten Quellcode auf zuvor bekannt gegebene Programmierkonventionen. Falls es in dieser Maßnahme zu einem Fehler kommt, erhält die Nutzerin eine genaue Fehlermeldung, wie und wo die Konvention verletzt wurde.

Da es in diesem Beispiel zwei Maßnahmen gab, die erfolgreich durchgeführt wurden, und eine Maßnahme, die Fehler entdeckt hat, lautet das Gesamtergebnis, dass Fehler entdeckt wurden. Das Gesamtergebnis wird in der oberen rechten Ecke angezeigt.

7.5 Benutzung des ASB-Systems aus Dozentensicht

Wie Studierende können sich Dozenten mittels ihrer Kennung der Hochschule Trier am ASB-System anmelden. Nach erfolgreicher Anmeldung sind diese in der Lage, ihre im ASB hinterlegten Lehrveranstaltungen zu administrieren. Dazu gehört das Anlegen von Aufgaben innerhalb ihrer Lehrveranstaltungen. Diese bestehen neben der Beschreibung und Abgabefrist aus einer oder mehreren Bewertungsmaßnahmen. Auf den Aufbau von Bewertungsmaßnahmen wird in Kapitel 16 eingegangen.

Wichtigster Bestandteil der Nutzung des ASB-Systems aus Dozentensicht ist die Bewertungsübersicht einer Aufgabe. Hier liegt dem Dozenten eine vollständige Übersicht aller Bewertungen der Studierenden vor, die Lösungen zu einer bestimmten Aufgabe eingereicht haben. Durch Auswählen eines einzelnen Studierenden wird eine wie schon in Abschnitt 7.4 beschriebene detaillierte Bewertungsübersicht angezeigt. Zusätzlich zu den Bewertungen kann der Dozent den hochgeladenen Quelltext im Browser betrachten und auch herunterladen. Dies unterstützt die Lehrenden in der manuellen Überprüfung studentischer Lösungen. Abbildung 7.3 zeigt, dass zunächst die Postfix-Aufgabe ausgewählt wurde. Links unten sieht man eine Übersicht über alle Studierende, die zu der dazugehörigen

² <http://checkstyle.sourceforge.net>

Lehrveranstaltung registriert sind. Man kann dadurch auch leicht erkennen, wer nichts eingereicht hat. Wird eine Person aus der Liste ausgewählt, so werden die dazugehörigen Bewertungsergebnisse angezeigt. Neben den für Studierende einsehbaren Bewertungsmaßnahmen kann das ASB-System auch versteckte Maßnahmen durchführen, deren Resultate nur die Dozenten sehen können. Hierzu zählt beispielsweise die selbst entwickelte Plagiatserkennung *JMaat*. Diese wird nach Ablauf der Abgabefrist angestoßen und zeigt dem Dozenten eine Übersicht möglicher Plagiate.

7.6 Vorgaben für ASB-Aufgaben

Zur Bewertung von Lösungen müssen bestimmte Vorgaben seitens der Studierenden eingehalten werden. Diese werden zu den jeweiligen ASB-Aufgaben auf den Übungsblättern erläutert. So werden in vielen Aufgaben Package-, Klassen- und Methodennamen vorgegeben. Darüberhinaus ist es notwendig, dass für manche Methoden die Parametertypen und der Rückgabotyp genau festgelegt werden. Dies ist notwendig, damit die durchzuführenden Unit-Tests ausgeführt werden können. Häufig werden hierfür vordefinierte Schnittstellen in Form von Dateien zur Verfügung gestellt, die die Studierenden implementieren müssen. Den Unit-

The screenshot displays the ASB system's evaluation overview. At the top, there are navigation tabs: 'Übersicht', 'Lehrveranstaltungen', 'Lösungsbearbeitung', and 'Benutzerkonto'. Below this, there are filters for 'Nach Lehrveranstaltung filtern', 'Nach Übungsblatt filtern', and 'Nach Status filtern'. A table lists assignments with columns: 'Lehrveranstaltung', 'Übungsblatt', 'Aufgabe', 'Phase', 'Typ', and 'Status'. The table contains five rows of data. Below the table, there are buttons for 'Gesamtübersicht anzeigen', 'Alle Lösungen herunterladen', and 'Herunterladen'. A search bar shows 'Lösung: postfix.zip'. On the left, there is a list of users: Britta Herres, Susanne Muster, Max Mustermann, and David Schuster, each with a status icon. The main content area shows a 'Detailliertes Bewertungsergebnis' for the task 'PostFix-Rechner', with sub-items: 'Ergebnisse des Compilerlaufs', 'Layouttests', and 'Funktionalitätstests für die Klasse PostFixCalculator'.

Lehrveranstaltung	Übungsblatt	Aufgabe	Phase	Typ	Status
Entwicklung mobiler Anwendungen (E...	Activities	CountryInfo	1		Geschlossen
Entwicklung mobiler Anwendungen (E...	Einführung	PlusMinus	1		Geschlossen
Entwicklung mobiler Anwendungen (E...	PostFix	PostFix-Rechner	1		Offen
Entwicklung mobiler Anwendungen (E...	Services	Counter	1		Geschlossen
Entwicklung mobiler Anwendungen (E...	UI-Elements	ButtonView	1		Geschlossen

Abbildung 7.3: Bewertungsübersicht eines Dozenten zu einer Aufgabe am ASB-System

Tests sind größtenteils Signaturprüfungen vorgeschaltet, die überprüfen, ob die Lösung diese Vorgaben erfüllt. Ist dies nicht der Fall, werden keine Funktionalitätstests angestoßen, sondern es erfolgt eine Fehlermeldung, dass die eingereichte Klasse zum Beispiel eine bestimmte Methode nicht besitzt.

Zum Simulieren von Nutzertests im Kontext der Programmierung von Android-Apps und JavaFX-Programmen müssen die Interaktionselemente, die in den Programmen der Studierenden erzeugt werden, mit vorgegebenen Identifikatoren ausgestattet werden. Dies gestattet eine automatisierte Prüfung der erstellten Benutzeroberflächen, indem die studentischen Programme gestartet, mittels der Identifikatoren die Interaktionselemente identifiziert, Nutzerinteraktionen wie Eingaben in Textfelder oder Klicks auf Buttons simuliert und daraus resultierende Anzeigen des Programms auf Richtigkeit geprüft werden.

Durch Betrachtung von Abbildung 7.4 erhält man eine gewisse Vorstellung, wie solche Tests implementiert werden. Es geht dabei um das Testen der Implementierung eines Postfix-Rechners als Android-App. Der erste Teil des dargestellten Codes besteht aus der Simulation einer Benutzereingabe. In diesem Fall drückt man auf den Button mit der Ziffer 4 und anschließend auf den Push-Button (damit ist die Eingabe einer Zahl beendet, die Zahl 4 wird somit auf den Keller gelegt). Danach wird der Button mit der Ziffer 3 gedrückt und wieder Push (damit wird nun auch noch die Zahl 3 auf den Keller gelegt). Schließlich wird der Plus-Button gedrückt. Damit ist die Simulation der Benutzereingabe beendet. Wichtig ist hierbei, dass die Studierenden die in der Aufgabenstellung vorgeschriebenen Identifikatoren für ihre Buttons wie `buttonDigit4`, `buttonPush` und `buttonOperationPlus` genauso einhalten, denn ansonsten kann der Test nicht durchgeführt werden. Wenn die App nach dieser simulierten Eingabe richtig implementiert wurde, müsste sie

```
/** Test Plus */
@TestDestruction(R.string.FuncTestAdd)
public void testPlus() {

    TouchUtils.clickView(this, this.buttonDigit4);
    TouchUtils.clickView(this, this.buttonPush);

    TouchUtils.clickView(this, this.buttonDigit3);
    TouchUtils.clickView(this, this.buttonPush);

    TouchUtils.clickView(this, this.buttonOperationPlus);

    final float result = this.readTextViewValue();

    Assert.assertEquals(this.res.getString(R.string.FuncTestAddMsg1, result), 7F, result, 0.001F);
}
```

Abbildung 7.4: Programmcode eines Tests für eine Android-App

jetzt die beiden obersten Kellerelemente (3 und 4) vom Keller genommen, addiert und das Ergebnis als oberstes Element auf den Keller gelegt und angezeigt haben. In der vorletzten Zeile wird der angezeigte Wert des obersten Kellerelements von der Oberfläche ausgelesen. Dazu wird eine Hilfsmethode verwendet. In der letzten Zeile schließlich wird der ausgelesene Wert mit 7 (dem erwarteten Resultat der Addition von 4 und 3) verglichen. Sind beide Werte (bis auf eine angegebene Toleranz von 0.001) gleich, endet der Test erfolgreich, andernfalls mit einem Fehler.

Das Einreichen von Lösungen wird den Studierenden zu Beginn der Lehrveranstaltungen erläutert. Das ASB-System erwartet alle zu testenden und darüber hinaus genutzten Dateien (*.java, *.fxml, *.xml, ...) innerhalb der vorgegebenen Paketstruktur als ZIP-Archiv. Dieses wird auf dem ASB-System entpackt und den Bewertungsmaßnahmen zur Bewertung übergeben.

Die Tests zu den Lehrveranstaltungen werden von Mitarbeiterinnen und Mitarbeitern des Fachbereiches in Git-Repositories gepflegt und erweitert. Sowohl das ASB-System selbst als auch die Tests sind Eigenentwicklungen der Hochschule Trier. Die Testkonzepte zum Testen von Android-Applikationen sind im Rahmen von studentischen Projekten entstanden.

7.7 Erfahrungen

Die Erfahrung zeigt, dass das ASB-System im Allgemeinen sehr gut angenommen und von vielen Studierenden sehr positiv hervorgehoben wird. In den hochschulweit standardisierten Fragebögen zur Lehrveranstaltungsbewertung gibt es (selbstverständlich) keine Frage zum ASB-System. In den Freitextantworten wird die Nutzung des ASB-Systems immer wieder positiv hervorgehoben, insbesondere auch von den Fernstudierenden. Es gibt jedoch auch Situationen, bei denen die Studierenden sich über das ASB-System beklagen. Eine typische Aussage lautet dann: *„Ich versuche jetzt seit Stunden, die vom ASB-Sever gemeldeten Fehler zu beheben. Bei mir lokal auf dem Rechner funktioniert alles einwandfrei, aber der ASB-Server meldet ständig Fehler. Ich drehe gleich durch“*. Die Ursache dafür liegt nach unserer Erfahrung nur in einigen solcher Fälle beim ASB-System. So kann es beispielsweise vorkommen, dass die Aufgabenspezifikation und die Tests nicht konsistent zueinander sind, insbesondere wenn die Aufgabenbeschreibung angepasst und dabei vergessen wird, diese Änderungen auch in den automatischen Tests nachzuvollziehen. Dann steigt die Anzahl der Forumsbeiträge auf der Lernplattform *Stud.IP* stark an. In der Regel werden die Tests oder die Aufgabenstellung dann umgehend angepasst und die Änderung im Forum mitgeteilt. In

manchen Fällen finden die Studierenden durch Ausprobieren heraus, was getan werden muss, damit die Lösung vom ASB-Server nicht mehr beanstandet wird, und teilen dies den anderen Studierenden im Forum mit. In seltenen Fällen treten auch noch Fehler im ASB-System selbst auf, was zu einer gewissen Frustration führt. Oft liegt die Ursache des Problems allerdings nicht auf Seiten des ASB-Servers, sondern auf Seiten der Studierenden selbst. Die Studierenden probieren ihre Software aus, kommen dabei jedoch nicht in solche Situationen, die in den Tests auf dem ASB-System überprüft werden. Wenn dann die Studierenden die Fehlermeldungen des ASB-Systems nicht verstehen oder nicht richtig interpretieren, was manchmal an zu knappen oder nicht treffend formulierten Meldungen des ASB-Systems liegen kann, dann beklagen sich die Studierenden über das Forum oder per Mail. Die Lehrenden geben daraufhin dezente Hinweise, in welche Richtung die eingereichte Lösung geändert werden sollte.

Wie soeben angesprochen können schwer verständliche Fehlermeldungen ein Problem sein. Sind hingegen die Fehlermeldungen sehr klar und ausführlich formuliert, so kann dies wiederum auch zu Problemen führen. So haben wir zum Beispiel schon beobachtet, dass die Studierenden ihre Programme genau auf die getesteten Spezialfälle hin ausrichten, wenn der ASB-Server sehr genau meldet, was das Programm ausgibt und was erwartet wird. Um zu verstehen was damit gemeint ist, stelle man sich als Beispiel ein Programm vor, das zwei eingegebene Zahlen addieren und das Ergebnis ausgeben soll. In einem Fehlerfall würde das ASB-System beispielsweise melden: *„Es wurden die Zahlen 3 und 4 eingegeben. Ihr Programm gibt 9 aus, erwartet wird aber 7“*. Wir haben schon erlebt, dass dann Programme so geändert werden, dass die Eingaben ignoriert werden und genau das vom ASB-System erforderte Resultat produziert wird. Würde bezogen auf das oben angegebene Beispiel immer 7 als Ergebnis der Addition ausgegeben, wäre der Test zwar bestanden, die programmierte Lösung wäre aber eigentlich falsch. Als Gegenmaßnahmen führen wir nach Möglichkeit mehrere unterschiedliche Tests aus. Wenn es die Aufgabenstellung erlaubt, variieren die Tests die Eingaben auch zufällig, sodass bei jedem erneuten Hochladen einer Lösung ein anderer Testfall geprüft wird. Grundsätzlich geben wir den Testcode den Studierenden nicht bekannt.

Das ASB-System wird von den Lehrenden ausschließlich als Assistenzsystem genutzt. Es geht dabei um die Vergabe des Nachweises einer unbenoteten Prüfungsvorleistung. Zur Unterstützung von Prüfungen wird das System nicht eingesetzt (wie zuvor beschrieben wurde, gibt es auch kein Punktesystem). Und auch die Bestätigung der erbrachten Prüfungsvorleistung erfolgt nicht vollautomatisch. Zum Erwerb der Prüfungsvorleistung wird nicht gefordert, dass ein bestimmter Prozentsatz der von den Studierenden eingereichten Lösungen fehlerfrei ist; es

wird lediglich gefordert, dass es das „ernsthafte Bemühen“ gibt, einen bestimmten Teil der Aufgaben zu lösen. Das ASB-System unterstützt die Lehrenden dabei herauszufinden, welche Lösungen sie genauer betrachten sollten, um dann zu entscheiden, ob ein „ernsthaftes Bemühen“ vorliegt. Eine eingereichte Lösung wird auf jeden Fall betrachtet, wenn dazu eine Frage oder Beschwerde zum Verhalten des ASB-Systems eintrifft. Aber auch ohne eine solche Rückmeldung ist ein Blick in fehlerhaft gekennzeichnete Lösungen interessant. Man kann dann nicht nur das „ernsthafte Bemühen“ überprüfen, sondern in den Übungsstunden ganz speziell darauf hinweisen, wie etwas nicht gelöst werden sollte. Stichprobenhaft werden auch als korrekt markierte Lösungen aufgrund der oben beschriebenen Tricksereien angeschaut. Dies ist insbesondere dann nötig, wenn es zu einer ASB-Aufgabe keine Tests gibt und nur formale Überprüfungen (korrekte Syntax, Einhaltung von Codierungskonventionen) durchgeführt werden. Es ist sogar schon vorgekommen, dass Programme, die mit der gestellten Aufgabe nichts zu tun haben, eingereicht wurden (z. B. Lösungen zu anderen Aufgaben). In diesem Fall werden die Studierenden ermahnt, dass dies als Täuschungsversuch interpretiert und im wiederholten Fall die Prüfungsvorleistung nicht vergeben wird. Ein weiteres Problem ist die einfache Kopierbarkeit von Dateien. Musterlösungen zu den Aufgaben werden seit langer Zeit nicht mehr herausgegeben, weil dies in der Vergangenheit zu dem Problem geführt hat, dass von einigen Studierenden die Musterlösung des Vorjahres inklusive aller Kommentare eingereicht wird. Auch Plagiate sind ein Problem. Die Studierenden werden intensiv darauf hingewiesen, dass sie alle ihre Lösungen selbst zu programmieren haben. „Gruppenarbeit“ ist grundsätzlich nicht erlaubt, da die Lehrveranstaltungen auf den Erwerb individueller Programmierkompetenz ausgerichtet sind, was nicht immer von allen Studierenden klaglos akzeptiert wird. Ob Lösungen als Plagiate eingestuft werden, wird ebenfalls nicht vollautomatisch allein aufgrund der ASB-Meldungen entschieden. Die im ASB-System genutzte Plagiatserkennung gibt lediglich Hinweise, welche Lösungen sehr ähnlich sind. Diese verdächtigen Lösungen werden dann von den Lehrenden näher in Augenschein genommen. Dies führt bei starkem Verdacht zu Verwarnungen, in besonders dreisten Fällen direkt zum Verlust der Prüfungsvorleistung. Aus unserer Erfahrung scheinen solche abschreckenden Vorgehensweisen notwendig zu sein.

Studierenden, die zwar eine korrekte Lösung eingereicht haben, die sich jedoch darüber hinaus fragen, ob ihre Lösung „Schönheitsmängel“ besitzt und trotz erfolgreicher ASB-Prüfung verbessert werden kann, bieten wir seit Kurzem ein sogenanntes Programmierberatungsbüro an, in dem Assistenten und Assistentinnen den Studierenden Verbesserungshinweise zu ihrer Software geben.

7.8 Fazit und Ausblick

Insbesondere in Zeiten mit hohen Studierendenzahlen bietet das ASB-System eine wertvolle und komfortable Unterstützung bei der Durchführung des Übungsbetriebs programmierlastiger Lehrveranstaltungen im Fachbereich Informatik der Hochschule Trier. Bei der Nutzung des ASB-Systems fallen immer wieder Punkte auf, wie das ASB-System, insbesondere dessen Bedienbarkeit, verbessert werden könnte. Je nach Aufwand werden dann entsprechende Änderungen am ASB-System vorgenommen oder auch nicht.

Eine signifikante Erhöhung der Automatisierung bei der Bewertung studentischer Lösungen ist dagegen im Augenblick schwer vorstellbar. Die Lehrenden wissen, dass beim Programmieren aufgrund eines kleinen Fehlers sehr viel schief gehen kann. Zum Beispiel kann ein vergessenes Semikolon einen Syntaxfehler erzeugen, sodass die studentischen Programme nicht übersetzt, damit nicht ausgeführt und nicht getestet werden können, oder eine fehlende Initialisierung kann in einer `NullPointerException` resultieren, die alle Testläufe scheitern lässt. Würde man in solchen Fällen vollautomatisch zu dem Ergebnis kommen, dass nichts an dem eingereichten Programm richtig ist, würde man die studentischen Programme sicher zu negativ bewerten. Auch die Überprüfung von Strukturvorgaben (z. B. ob ein Programm nach dem Model-View-Presenter-Prinzip gestaltet wurde) ist schwer automatisch überprüfbar. Umgekehrt werden auch die Studierenden immer wieder darauf hingewiesen, dass sie auf keinen Fall davon ausgehen dürfen, dass die Anzeige der Fehlerfreiheit durch ASB bedeutet, dass ihr Programm korrekt ist. Aus diesen Gründen wird das ASB-System nicht zur vollautomatischen Bewertung genutzt, sondern lediglich als Assistenzsystem sowohl für Lehrende als auch für Studierende.

Literatur für dieses Kapitel

- [Mor+07] Thiemo Morth u. a. „Automatische Bewertung studentischer Software“. In: *Workshop „Rechnerunterstütztes Selbststudium in der Informatik“*. Universität Siegen, 2007.